

- What were the decisions taken in the modelling?

Based on the database of the previous assignment, we converted the Entities into Collections and Merged the relational entities into the required collection (i.e book_category -> books collection). We also created the items collection to replace the relational entity book_order as this entity was integral to our solution.

The collections have a reference to another collection(ObjectId). We followed closely the previous model implementation.

- Why were these decisions taken?

We went with this approach because we thought it was the best way to translate our SQL model to a mongoDB model.

We focused on optimising the space and we decided to use collections references instead of lists of full objects. The decision was made after taking in consideration the schema design anti-patterns. We managed to avoid storing massive arrays and to not bloat the documents we came up with.

- What were the consequences of these decisions?

We didn't take advantage of the mongoDb flexibility to adjust the model to our needs. For example, querying the orders can be troublesome and slower as a result.

- What were the difficult and easy parts of the exercise?

The easy part was modelling the collections that don't have any references to other collections.

The hard part was figuring out how to make the connections between the more complicated collections (e.g. introducing items as a collection that connects books and order, instead of having an array of items in the order)

- How does that compare to relational databases?

MongoDb is more flexible when it comes to data modelling. It doesn't have to follow a strict set of rules to be implemented. It uses patterns and antipatterns to shape up the base of the model, the foundation, which can be further modified based on our needs.

- What are the advantages and disadvantages of MongoDB over relational databases for this exercise?

Advantages:

- Flexible - it can be designed based on the needs
- It doesn't enforce a strict schema and it can be expanded as the model evolves

- Fast

Disadvantages:

- Harder to query data across multiple/nested collections